

Ant Colony: A Community AI Experiment for Video Games

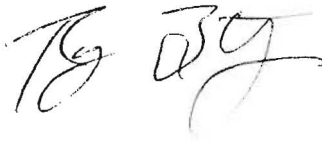
**An Honors Thesis (HONRS 499)**

**by**

Brandon Jones

**Thesis Advisor**

Ryan Bitzegaio

Two handwritten signatures in black ink. The first signature on the left is 'BJ' for Brandon Jones, and the second signature on the right is 'RB' for Ryan Bitzegaio.

**Ball State University  
Muncie, Indiana**

May 2012

**Expected Date of Graduation**

May 2012

## Abstract

Immersion is an increasingly large focus of developers in the media of video games. More than it ever was for games like *Chrono Trigger* on the Super Nintendo Entertainment System, static non-player characters following pre-defined paths and making the same exact decisions through each play through of the game can be a breakpoint for immersion. What modern games have lacked is a legitimate sense of being a character inside a dynamic community. To make an attempt at solving this problem, I created the game prototype *Ant Colony*. This prototype showcases a community with many personality types that make decisions independently, and based on certain factors. Each individual AI chooses where he or she wants to go in the world, and decides his or her own fates.

## Acknowledgements

I would like to thank Ryan Bitzegaio for advising me through this project. When my thoughts were too big, he helped me shrink them. I would never have finished this without his help.

I would like to thank Chris Turvey and Wayne Mock who helped me refine my initial proposal.

I would also like to acknowledge Cassie, Zach G., Zach D., Mike, and any of my other friends who I would talk to about this project to keep my sanity.

## Introduction

The video game industry has become a cultural juggernaut that eclipses most other industries in terms of revenue. I've played video games my entire life. Through the years, I've experienced and enjoyed every genre. When I was younger, my favorite genre was role-playing games (RPG), but unfortunately, that love diminished as I got older. For years I was frustrated, and I could not understand why I lacked the appreciation for the genre I once had. It certainly was not a lack of quality—the ratings for quality RPG titles had not dipped significantly.

When I played *Final Fantasy VII*, I kept getting frustrated with the characters in the environment. Destruction was occurring around them, and most of them did not run, and talking to them would always yield the same results. And yet, I gave an easy pass to characters reacting similarly in *Super Mario RPG: Legend of the Seven Stars* and *Chrono Trigger*. I have since diagnosed that I was less forgiving of the fault in *Final Fantasy* because (for its time) it was a very realistic looking game in 3D. Standards for a realistic appearance have since changed, but at the time it seemed egregiously unacceptable that the characters in a realistic environment did not behave realistically. By contrast at this time, characters in first person shooters (FPS) seemed to behave very realistically, because the only decisions they had to make were their own placement and shooting. The fact that artificial intelligence (AI) in these games would do something as simple as take cover when shot at seemed to be light years ahead of what was occurring in the RPG games that I had loved in the past. RPGs needed to pick up the pace, and I wondered for years how they could do it.

Recently, some bigger strides have been made. *The Elder Scrolls V: Skyrim*, by Bethesda Softworks, features a gigantic world for players to travel in, do quests, and save the world. The environments are lush, the characters in the world diverse, and as players go through this world

there's a real sense of progression and accomplishment. There's so much to do in the game that players tend not to realize how terrible the main quest line is because they become fully immersed in the world. Likewise, Bioware's *Mass Effect* series has increased player immersion by including a very cinematic and fully voiced conversation system. You see character reactions to statements, and have clear branches of how that can progress. But both of these games have a key flaw that hurts replay ability: they are highly-scripted experiences. The townspeople will be the same townspeople in every play-through of *Skyrim*, each making the same random greeting statement, and having the same basic conversation tree. Similarly, characters will appear in the same spaces and start with similar conversations through each play-through of the *Mass Effect* series, with the occasional minute change based on a past decision.

While these two are better examples of how RPGs can attain the same level of immersion as FPS games, they still aren't fully capitalizing on the idea of a dynamic world for players to participate in. Conversations are full of static branches; the killing of certain enemies fails to elicit a firm response from their allies or relatives; and characters follow the same pre-specified paths and places, and fail to make unique decisions in subsequent play-throughs. Nothing about the player's experience seems dynamic, which ruins the immersive quality of the games when players start again. It was this realization of the lack of dynamic communities in games that became the real genesis of my thesis project.

### Phase 1: Research and Planning

Video game AI was something I knew very little about, so the task of making a prototype that was heavily linked to video game AI principles was quite the undertaking. The first major phase of my project was spent just researching techniques used in the games industry, and any

other theory that I could get my hands on. I knew two basic things prior to the project: it needed to be easily managed, and it needed to process quickly.

Early on in my research I focused on trying to find tools that would satisfy this. Outside of gaming, functional or logical languages such as LISP or Prolog are commonly used for artificial intelligence. However these languages aren't commonly used in the video game industry, largely because their efficiency is for parsing information and decision trees, and they have virtually nothing in terms of support for graphical programming. Graphics being an ever-important portion of game development, I had to look elsewhere. Eventually I stumbled upon Stackless Python, which allowed game functions to be written in easily-maintained microthreads while still producing a huge speed increase over Python's standard distribution. This essentially works by maintaining smaller segments of code with very little overhead, and not outputting these instructions to the operating system's standard processing stack. *Eve Online* creators CCP Games Inc. use Stackless Python on their servers to maintain actions of their AI and users. (Jónsson, 2006) This proved quite scalable as they are able to maintain a population of more than 400,000 subscribers since the game launched in 2003. (Rubens, 2012)

With my desired programming language decided (or so I thought), I began examining game AI paradigms for my use. Fairly early on in my research, I recalled entity systems from my CS 315 class, and remembered that they were a pretty nice tool for splitting aspects of an object into interchangeable components. In essence an entity system is like building with Lego blocks. You can make whatever you want, as long as the blocks are of the right size so they can interconnect and work together correctly. Just because you use one block in one construct to represent one thing, it does not mean that you have to use the same exact block to accomplish a similar task in another construct. Instead you can change out something such as a personality

module when building similar constructs and you can recognize them for what they are by whether or not they have certain types of component parts. (Davies, 2011)

Finally, I had to decide just how I would map out the relationships of the community and design their decision models. Naturally, I wanted to strive for as realistic a model as possible, given that realism was the type of immersion I wanted to achieve with this thesis project. Among the resources I found, two specifically come to mind from my originally conceived prototype idea: Expectation-Managed Decision Making, and Luke Bergeron's Dynamic Conversation Engine. The Expectation-Managed Decision Making model gives each AI a mapped knowledge of its surroundings, and allows it to make decisions on what it expects should happen, as well as what is actually happening to its perceived knowledge. (Isla, 2005) The Dynamic Conversation Engine was more complex, but essentially an extension of the Expectation-Managed Decision Making model. Each AI would have an established and maintained knowledge of the world, its self, news, and of the conversation it's currently participating in. With each response it would update its knowledge, and react based on its known feelings of a situation. By implementing this knowledge, I could have each AI make a decision about where it wanted to go, and then allow it to follow that goal. (Bergeron, 2009)

## Phase 2: Shrinking the Scope

Game studios spend years with teams of talented programmers trying to implement ideas similar to mine. I came to realize as I was merely writing out the core design of this plot that I would never be able to finish such a project over the course of a semester. With help of my advisor, Ryan Bitzegaio, I paired down the idea into a much more manageable scope. Rather than a full and branching world, like I wanted, I would generate a much smaller community.

Also, I wasn't fully familiar with Python, and came to realize once I started trying to make test builds with it, that Stackless Python would not easily be packaged as an executable. This was counter-intuitive to how I wanted to deliver my overall project, so Stackless Python was scrapped in favor of Java. Finally, the systems that I was designing were too large for a one-man team to implement in one semester, and I definitely wanted to have a functioning prototype as my delivered content. So these systems were gutted and replaced with a system that would emulate their functionality as entity components, without requiring nearly as large an implementation overhead.

## Conclusion

The final game concept was *Ant Colony*. The premise: you play as an ant that has a premonition about impending doom for the ant colony, and in order to save the members of the colony, you have to talk to them and convince them to leave. The non-player characters were generated with procedurally assigned personality components that they would use to decide how they would respond to certain types of conversation responses. Each AI is also capable of deciding where in the colony that it would like to go to, and walking along a dynamically generated path decided by the A\* pathfinding algorithm, which takes a heuristic such as the distance from its desired goal and choosing only the possible positions that best fit that heuristic.

As a prototype it succeeds in showcasing the core of what I hoped to accomplish with my idea of a dynamic community. There are ants of varied personality types that make their own decisions, and take certain things like their relationships with other ants into account. The characters are different each play-through, and make dynamic decisions so the player has a unique experience every time he plays. While I did have to make several compromises from the

initial vision, the prototype is overall a successful example of how that vision could improve the immersive qualities of the RPGs that I used to love. By completing this project I have been able to showcase my skills as a developer, and I have expanded my knowledge into additional fields of programming. I am now more adept at programming artificial intelligence for video games, and can use these new skills in future endeavors as I pursue my development career.



### Works Cited

- Bergeron, L. (2009, July 25). *Dynamic Conversation Engine Concept*. Retrieved February 12, 2012, from Gamasutra:  
[http://www.gamasutra.com/blogs/LukeBergeron/20090725/85068/Dynamic\\_Conversation\\_Engine\\_Concept.php](http://www.gamasutra.com/blogs/LukeBergeron/20090725/85068/Dynamic_Conversation_Engine_Concept.php)
- Davies, T. (2011, January 23). *Entity Systems*. Retrieved February 13, 2012, from Tom Davies:  
<http://www.tomseysdavies.com/2011/01/23/entity-systems/>
- Isla, D. (2005, November 8). *Great Expectation-Formations*. Retrieved February 13, 2012, from AI-Blog: <http://www.ai-blog.net/archives/000104.html>
- Jónsson, K. V. (2006, February 25). Stackless Python in EVE. Dallas, TX, United States of America.
- Rubens, A. (2012, Mar 30). *Inside Eve Online's Fanfest 2012*. Retrieved May 1, 2012, from PC World: [http://www.pcworld.com/article/252940/inside\\_eve\\_onlines\\_fanfest\\_2012.html](http://www.pcworld.com/article/252940/inside_eve_onlines_fanfest_2012.html)